Fall 2014
SEI Research Review

Probabilistic Analysis of Time Sensitive Systems

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Jeffery Hansen

October 28, 2014

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **28 OCT 2014**	2. REPORT TYPE **N/A**	3. DATES COVERED

4. TITLE AND SUBTITLE **Probabilistic Analysis of Time Sensitive Systems**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Hansen /Jeffery**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES

The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**14**	

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0001787

What is the Problem?

Time-sensitive systems in uncertain environments have complex behaviors. How do we validate correct timing in such systems?

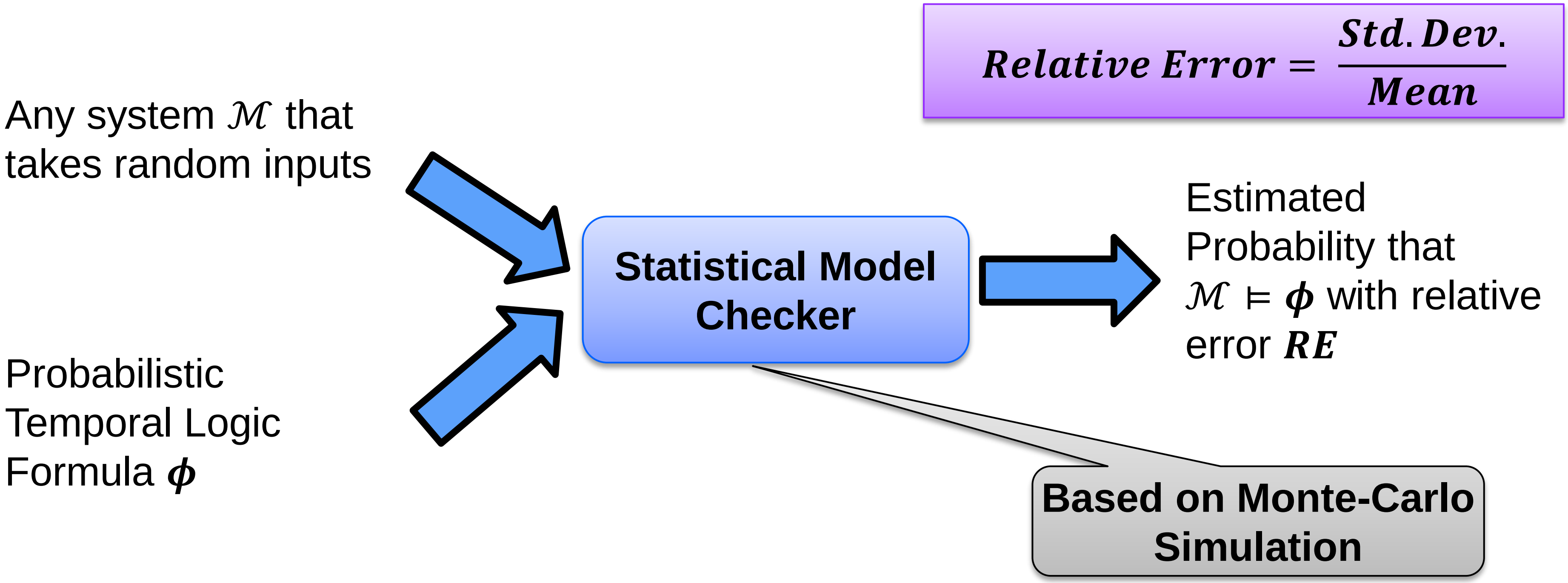
- Exact probabilistic verification is infeasible due to model size
- Black box testing does not yield bounded predictions
- Need formal approach for dealing with uncertainty
  - Accurate, bounded, probabilistic results
  - In reasonable time even for rare outcomes

Use statistical model checking to do a “smart sampling of the world”

- Simulation captures both random variables and timing (scheduling)
- Importance sampling “tilts” input distributions for efficient probability estimation of “rare” events

Note: We use “probability estimation” based statistical model checking. There is also a “hypothesis testing” based version.

Statistical Model Checking (SMC)



$$Relative\ Error = \frac{Std.Dev.}{Mean}$$

Any system $\mathcal{M}$ that takes random inputs

Probabilistic Temporal Logic Formula ϕ

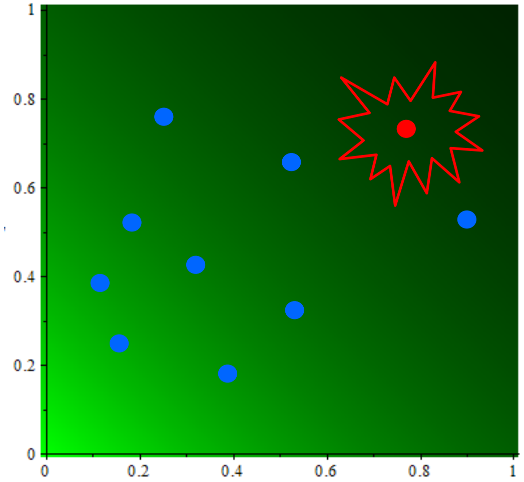
Statistical Model Checker

Estimated Probability that $\mathcal{M} \vDash \phi$ with relative error RE

Based on Monte-Carlo Simulation

- System properties described in formal language (UTSL, BLTL, etc.)
- Property is tested on "sample trajectories" (sequence of states)
- Each outcome can be treated as a Bernoulli random variable (i.e., coin flip)

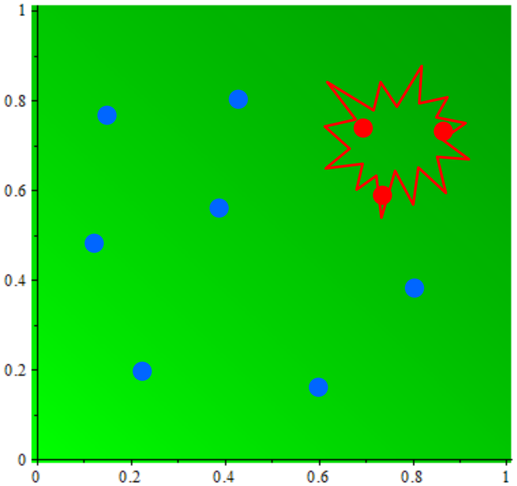
Probability Estimation with SMC



SMC Basics

- Indicator function $I(\vec{x}) = 1$ iff property holds for input $\vec{x}$.
- Relative Error $RE(\hat{p}) = \frac{\sqrt{var(\hat{p})}}{E[\hat{p}]}$ is measure of accuracy.
- Draw random samples from input distribution $f(\vec{x})$ until target Relative Error is met.
- Estimated probability that property holds is:

$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I(\vec{x}_i) = \frac{1}{10} = 0.1 \qquad RE(\hat{p}) = \frac{0.32}{0.1} = 3.2$$



Importance Sampling

- Modify input distribution to make rare properties more visible.
- Weighting function $W(\vec{x})$ maps solution back to original problem.
- Reduced relative error with same number of samples.

$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I(\vec{x}_i)W(\vec{x}_i) = \frac{0.2 + 0.5 + 0.3}{10} = 0.1$$

$$RE(\hat{p}) = \frac{0.18}{0.1} = 1.8$$

Osmosis SMC Tool

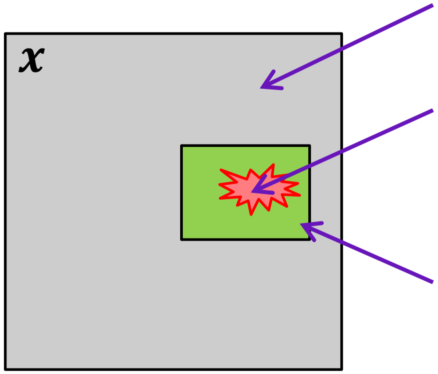
Osmosis is a tool for Statistical Model Checking (SMC) with Semantic Importance Sampling.

- Input model is written in subset of C.
- ASSERT() statements in model indicate conditions that must hold.
- Input probability distributions defined by the user.
- Osmosis returns the probability that at least one of the ASSERT() statements does not hold.
- Uses dReal[1] solver to build $I^*(\vec{x})$.
- Simulation halt condition based on:
  - Target relative error, or
  - Set number of simulations

[1] http://dreal.cs.cmu.edu/

Osmosis Main Algorithm

1. Generate approximation of fault region

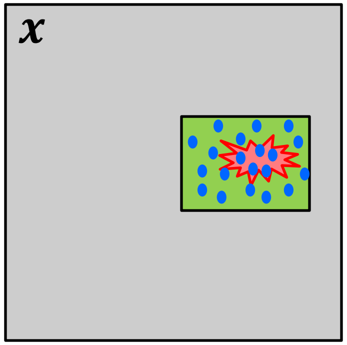


Input space

$\boldsymbol{I}(\vec{x})$: Indicator Function defines fault region

$\boldsymbol{I}^*(\vec{x})$: Abstract Indicator Function defines over-approximation

2. Conduct SMC and calc. failure prob.



a) $\hat{p}_{raw} = \frac{5 \longleftarrow \text{\# in Fault}}{20 \longleftarrow \text{Total \#}}$

b) $p^* = \frac{6}{64}$ Fraction of input $\boldsymbol{I}^*(\vec{x})$ covers

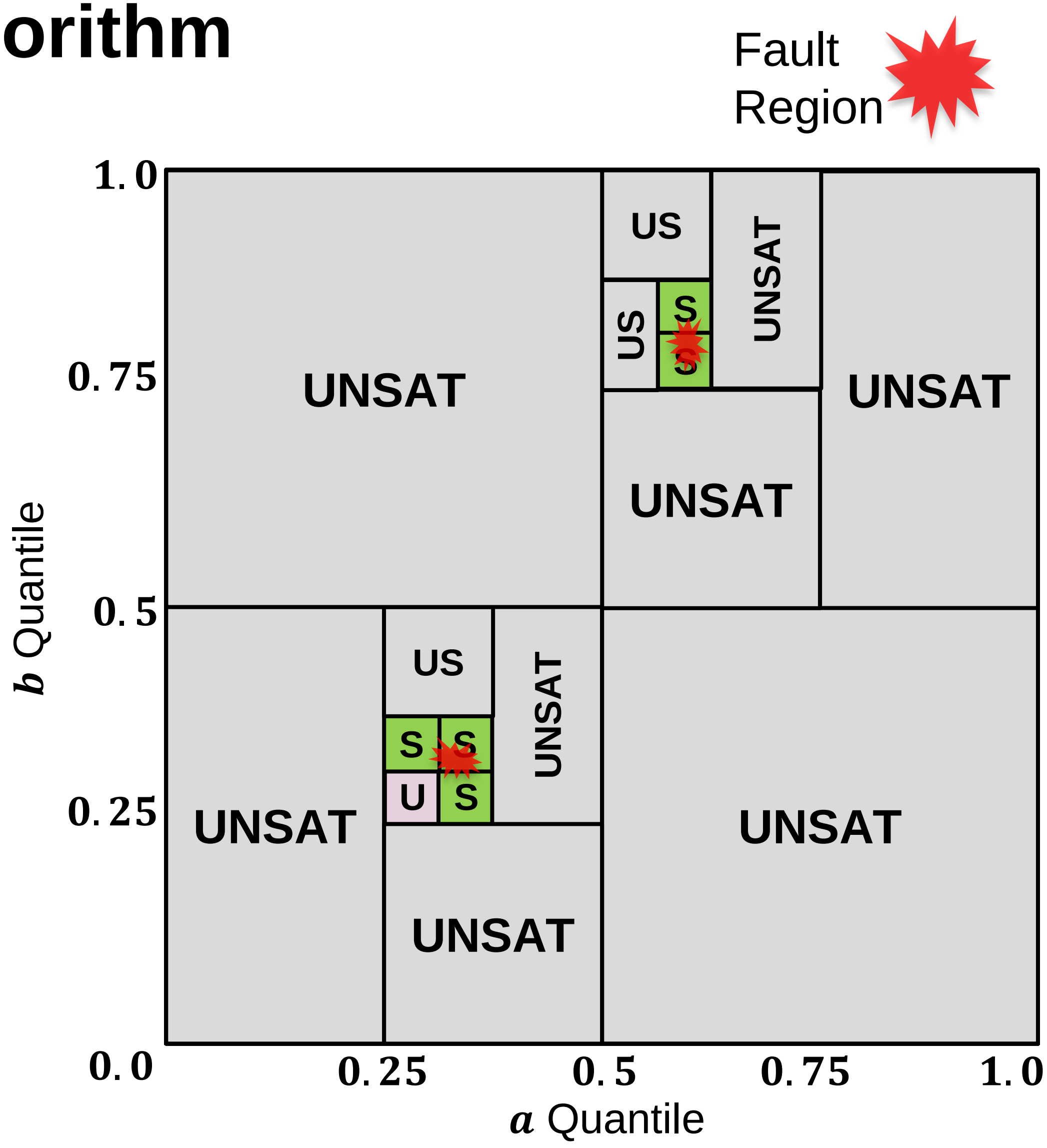
c) Failure prob. estimate is product of two values

$$\hat{p} = \hat{p}_{raw} p^* = 0.23$$

$I^*(\vec{x})$ Generation Algorithm

Algorithm:

1. Set the current "cube" as the full range of all inputs.
2. Apply dReal to the current cube.
3. If the result is "SAT", split cube into two equal probability cubes on one variable, and recursively apply at Step 2.



Example: Air Hockey Problem

Air Hockey Problem

- Table with a moving puck and a fixed target.
- Puck rebounds without friction.

Inputs

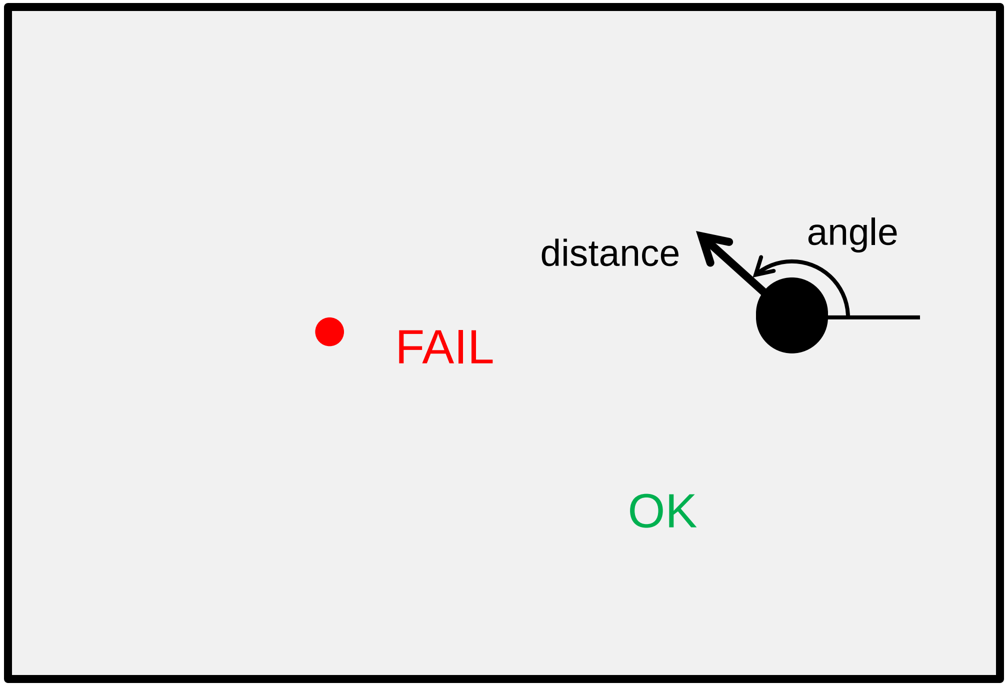
- Angle – Initial angle at which puck is hit.
- Distance – Total distance of travel for puck.

Failure Condition

- Puck stops on target (red dot).

Challenges

- Multiple failure areas in input space.
- Complex structure of failure area.

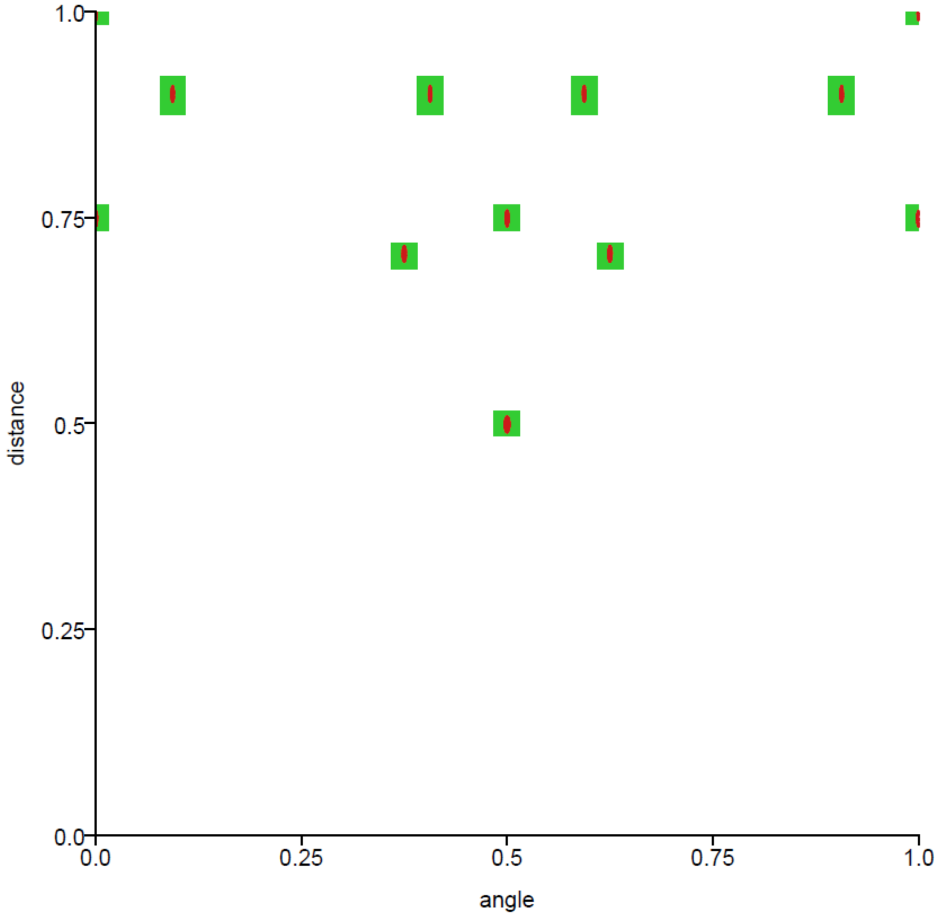


Fault Map for Air Hockey Problem

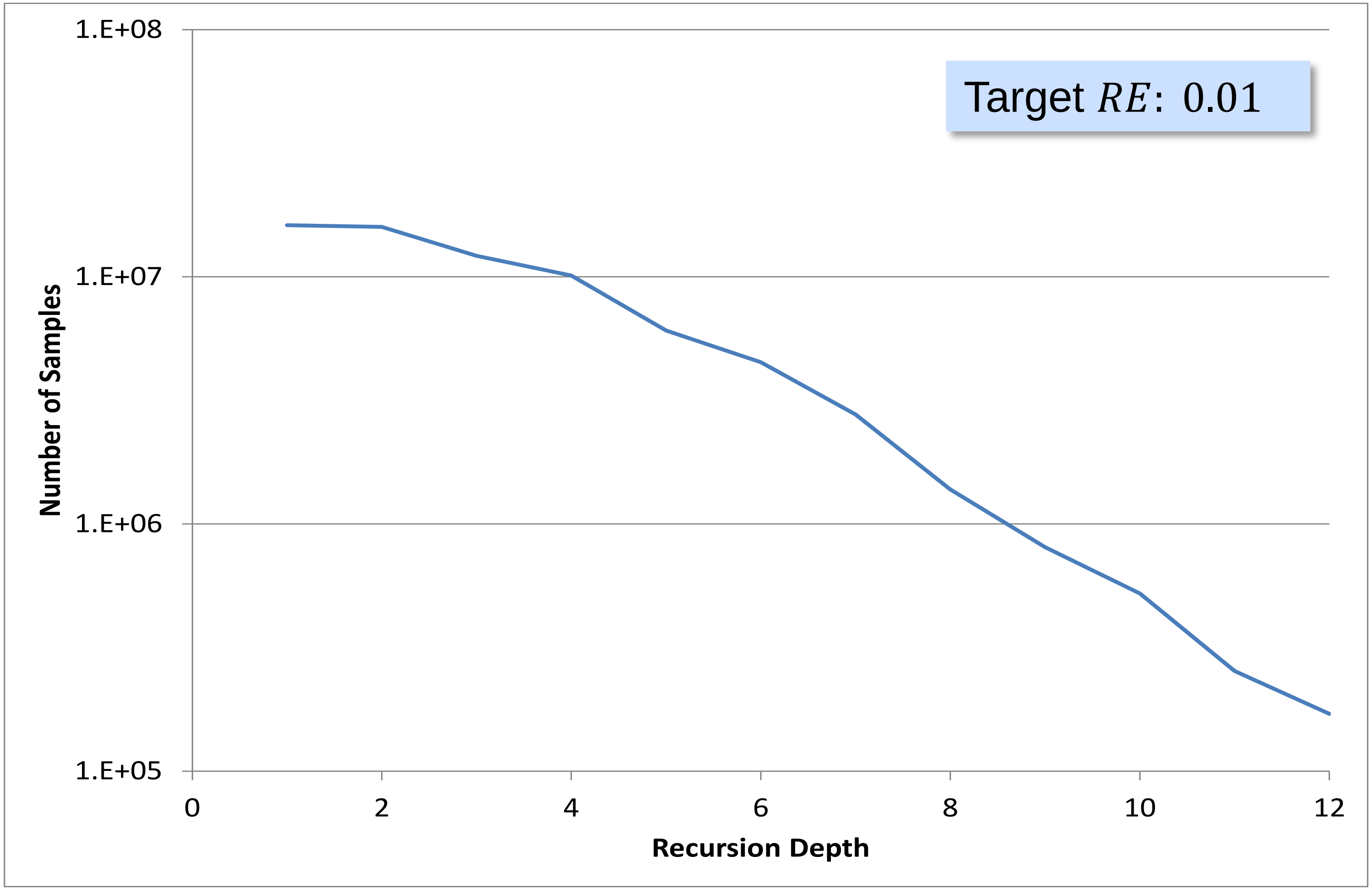
Fault map shows area of input space where faults are located.

- Plotted in CDF space.
- **Green** area indicates input space included in $I^*(x)$.
- **Red** area indicates input space include in $I(x)$.

Recursion Depth: 12

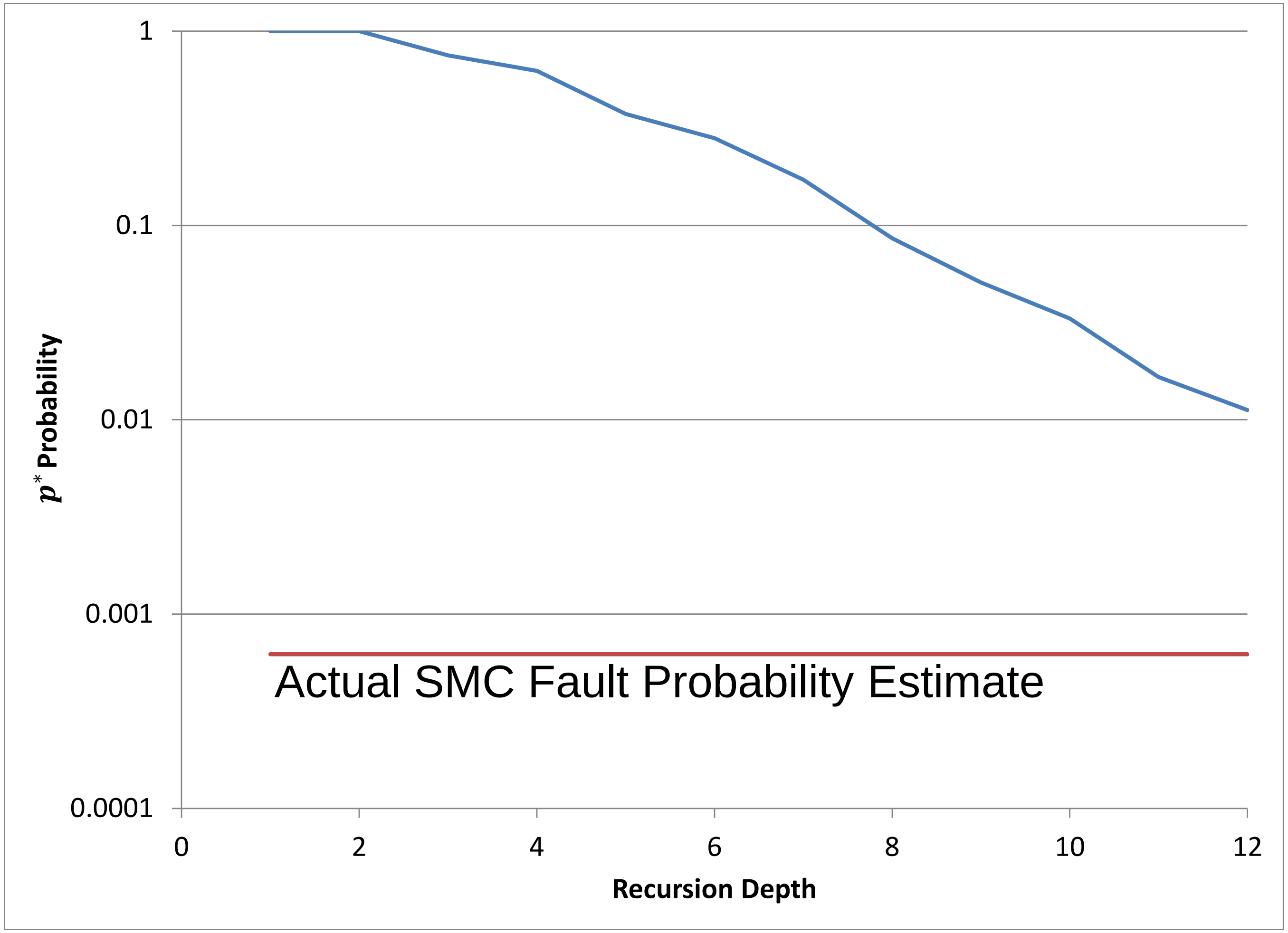


Sample Size vs Recursion Depth (Air Hockey)



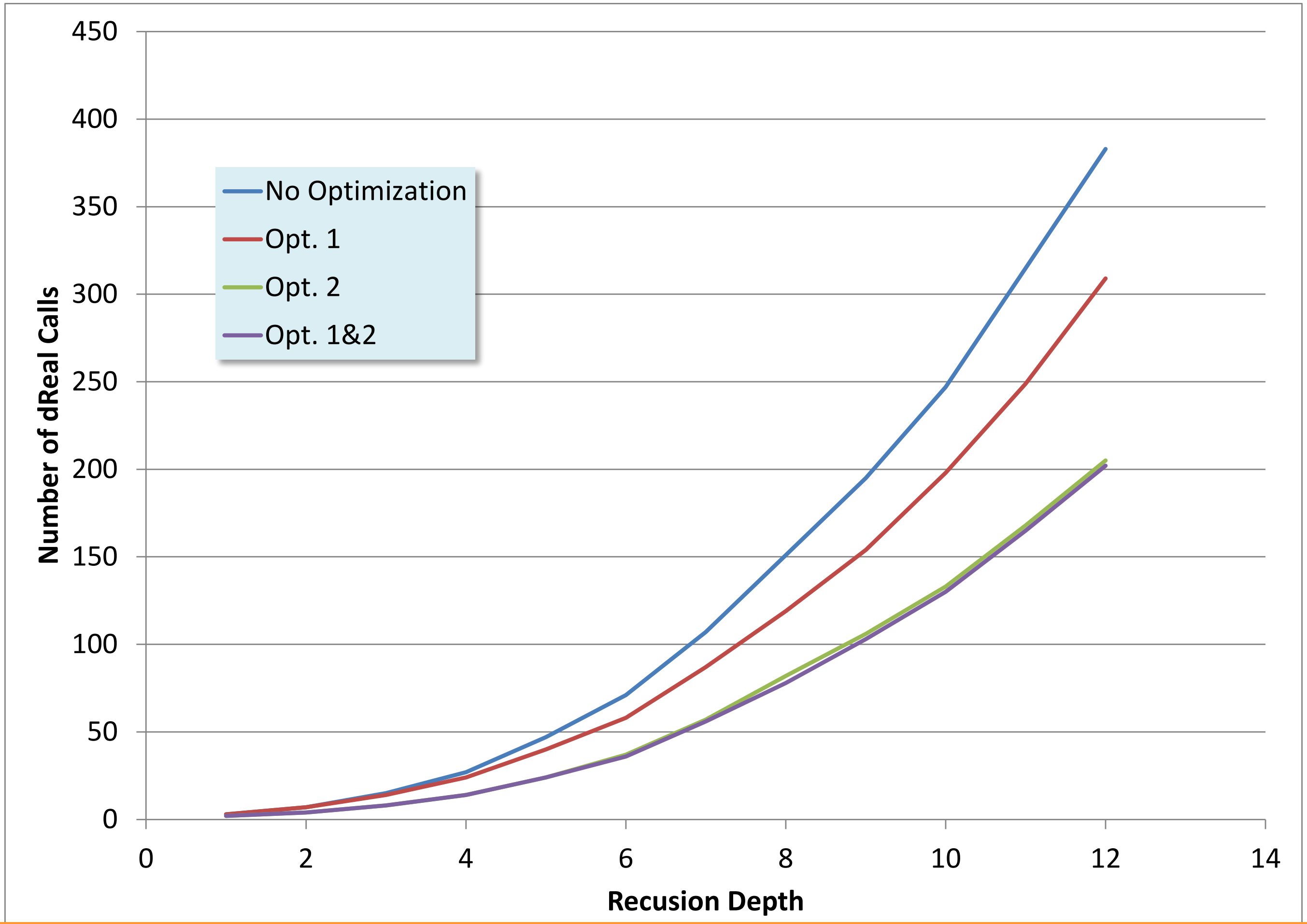
Simulation effort with SIS decreases exponentially with recursion depth.

p^* vs Recursion Depth (Air Hockey)

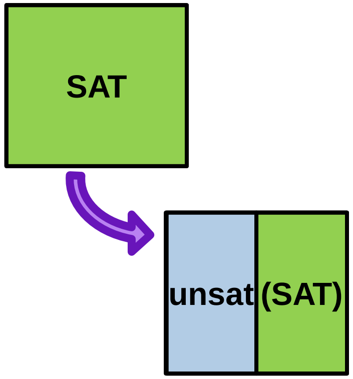


Upper-bound p^* becomes more accurate as recursion depth increases.

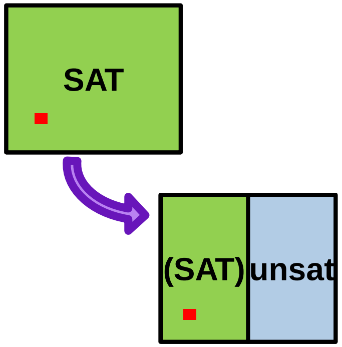
Effect of SIS Optimizations



Optimization 1
No call to dReal if first child call is unsat.



Optimization 2
Use counter-example from parent to avoid dReal calls on children.



Optimization 2 results in greatest benefit with factor of two reduction in number of calls to dReal. Small additional benefit by combining both methods.

Conclusion

Semantic Importance Sampling

- Create approximation of fault region using abstraction.
- Create an alternate input distribution for importance sampling.
- Level of approximation (recursion depth) is user tunable.
- Can reduce SMC sample size by orders of magnitude.

Osmosis tool

- Applies semantic importance sampling on a C-like specification.
- Uses the dReal SMT solver to build approximate fault region model.
- Can be applied when there are multiple fault regions.
- Optimization techniques can nearly halve number of dReal tests required.

Contact Information Slide Format

Presenter / Point of Contact

Jeffery Hansen, SSD
Telephone: +1 412-268-9565
Email: jhansen@sei.cmu.edu

U.S. Mail

Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web

www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations

Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257